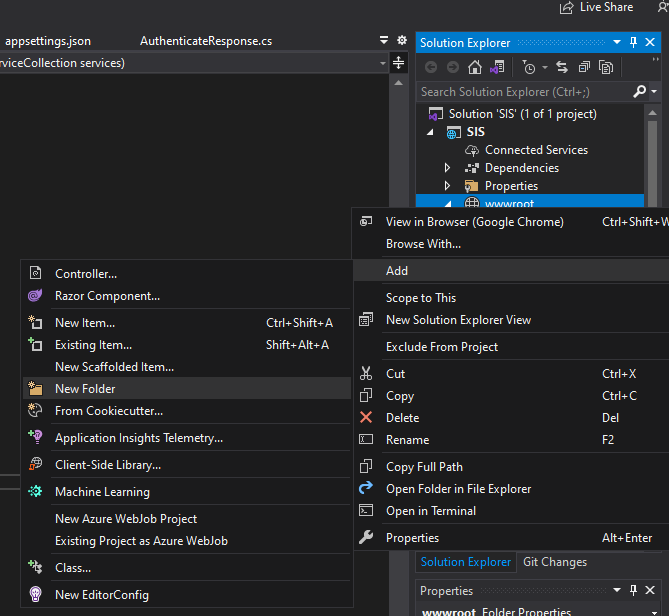
Activity 10

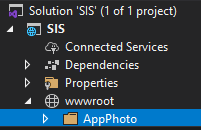
Download **Activity10.zip**

Download **Dropzone.zip**

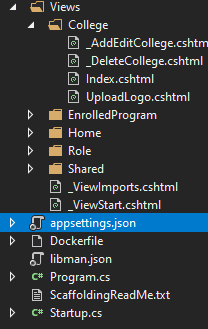
Step 1. Open your SIS Project

Step 2. On your solution Explorer, go to wwwroot add **AppPhoto** Folder





Step 3. Go to AppSettings add the following script

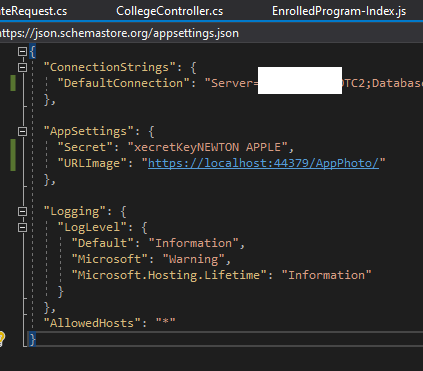


"AppSettings": {

"Secret": "xecretKeyNEWTON APPLE",

"URLImage": "https://localhost:44379/AppPhoto/"

},



Step 4. Go to Startup.cs add the following code inside **public void ConfigureServices(IServiceCollection services)**

var appSettingsSection = Configuration.GetSection("AppSettings");

services.Configure<AppSettings>(appSettingsSection);

// configure jwt authentication

var appSettings = appSettingsSection.Get<AppSettings>();

var key = Encoding.ASCII.GetBytes(appSettings.Secret);

services.AddTransient<IUserService, UserService>();

services.AddScoped<IUserService, UserService>();

services.AddTransient<ICollegeService, CollegeService>();

services.AddScoped<ICollegeService, CollegeService>();

services.AddSingleton<IFileProvider>(

new PhysicalFileProvider(Path.Combine(Directory.GetCurrentDirectory(), "wwwroot/AppPhoto")));

Step 5. Add Folder **Helper and Class AppSettings**



Copy this Code.

namespace WebApi.Helpers

{

public class AppSettings

{

public string Secret { get; set; }

public string URLImage { get; set; }

}

}

Step 6. Add Folder **Entities** and Class **RefreshToken**.cs copy the following code.

using System;

using System.ComponentModel;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Text.Json.Serialization;

using Microsoft.EntityFrameworkCore;

namespace WebApi.Entities

{

[Owned]

public class RefreshToken

{

[Key]

[JsonIgnore]

public int Id { get; set; }

public string Token { get; set; }

public DateTime Expires { get; set; }

public bool IsExpired => DateTime.UtcNow >= Expires;

public DateTime Created { get; set; }

public string CreatedByIp { get; set; }

public DateTime? Revoked { get; set; }

public string RevokedByIp { get; set; }

public string ReplacedByToken { get; set; }

public bool IsActive => Revoked == null && !IsExpired;

}

}

Step 7. On **Model** Folder add Class **UserToken** copy the following code.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Linq;

using System.Text.Json.Serialization;

using System.Threading.Tasks;

using WebApi.Entities;

namespace SIS.Models

{

public class UserToken

{

[DisplayName("TokenID")]

[Key]

[DatabaseGenerated(DatabaseGeneratedOption.Identity)]

public int Id { get; set; }

public string Username { get; set; }

[JsonIgnore]

public List<RefreshToken> RefreshTokens { get; set; }

}

}

Step 8. On Model add Class **AuthenticateRequest** copy the following code.

using System.ComponentModel.DataAnnotations;

namespace WebApi.Models

{

public class AuthenticateRequest

{

[Required]

public string Username { get; set; }

[Required]

public string Password { get; set; }

}

}

Step 9. On **Model** Folder add Class **AuthenticateResponse** copy the following code.

using SIS.Models;

using System.Text.Json.Serialization;

using WebApi.Entities;

namespace WebApi.Models

{

public class AuthenticateResponse

{

public int Id { get; set; }

public string Username { get; set; }

public string JwtToken { get; set; }

[JsonIgnore] // refresh token is returned in http only cookie

public string RefreshToken { get; set; }

public AuthenticateResponse(UserToken user, string jwtToken, string refreshToken)

{

Id = user.Id;

Username = user.Username;

JwtToken = jwtToken;

RefreshToken = refreshToken;

}

}

}

Step 10. On **Model** Folder add Class **RevokeTokenRequest** copy the following code.

namespace WebApi.Models

{

public class RevokeTokenRequest

{

public string Token { get; set; }

}

}

Step 11. On **Model** Folder update Class **College** copy the following code.

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc.Rendering;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Linq;

using System.Threading.Tasks;

namespace SIS.Models

{

public class College

{

[DisplayName("College ID")]

[Key]

[DatabaseGenerated(DatabaseGeneratedOption.Identity)]

public int CollegeID { get; set; }

[Column(TypeName = "char(10)")]

[DisplayName("College Code")]

public string CollegeCode { get; set; }

[Column(TypeName = "varchar(50)")]

[DisplayName("College Name")]

public string CollegeName { get; set; }

[Column(TypeName = "varchar(50)")]

[DisplayName("Dean")]

public string NameofDean { get; set; }

public bool IsActive { get; set; }

[DisplayName("Date Recognize")]

public DateTime RecognizeDate { get; set; }

[DisplayName("No. of Prog.")]

public int NumberOfProgram { get; set; }

[Column(TypeName = "varchar(100)")]

[DisplayName("College Logo")]

public string Logo { get; set; }

[NotMapped]

public string LogoImageUrl { get; set; }

[Column(TypeName = "char(36)")]

[DisplayName("FileStamp")]

public string FileStamp { get; set; }

[NotMapped]

public IFormFile UploadFiles { set; get; }

}

}

Step 11. On **Data** Folder add Class **ApplicationDbContext** copy the following code.

public DbSet<UserToken> UserTokens { get; set; }

run the following after

Add-Migration userTokenKey

Update-Database

Step 12. Add **Service** Folder add Class **CollegeService** copy inside the CollegeService.txt file

Step 13. On **Service** Folder add Class **UserService** copy inside the UserService.txt file

Step 14. On **Controller** Folder add Controller **APIController** copy inside the APIController.txt file

Step 15. On **Controller** Folder add Controller **UsersController** copy inside the UsersController.txt file

Step 16. On **Controller** Folder update Controller **CollegeController** copy inside the CollegeController.txt file

Step 17. Copy Views/College to your solution, you replace all the chtml in your current Project.

Step 18. Extract dropzone.zip and copy the entire folder www folder